

# Poster: Delay Tolerant Networking for Sensor Networks

Melissa Ho, Kevin Fall  
{melissa.r.ho, kfall}@intel.com  
Intel Research Berkeley

## ABSTRACT

Sensor network deployments may be far removed from communications infrastructure such as the Internet. Yet, to be maximally useful, these networks of sensors must ultimately be connected to data storage and analysis facilities. Providing connectivity for such networks may involve exotic and unusual methods of data transfer. In addition, within such networks, problems of intermittent connectivity due to power scheduling, node failure, and packet losses from unpredictable external factors are frequently encountered. As a solution the first problem, we propose the use of the Delay Tolerant Networking (DTN) architecture, which provides reliable data communication across heterogeneous, failure-prone networks. For the second problem, we show elements of the DTN architecture can be employed within a sensor network to mitigate communication interruptions. We suggest an overall architecture, employing the full DTN architecture for access to sensor networks, combined with a subset implementation of its design for use within sensor networks, as a reasonable basis for a complete delay-tolerant sensor network architecture.

## 1. INTRODUCTION

Environmental monitoring is one of several applications utilizing networks of sensors. While some environments may be easily equipped with the infrastructure required to return data from a network to the operator's network, so-called *reachback* infrastructure may be too risky, difficult, or expensive to install. In such cases, alternatives must be investigated to achieve data-return. Such alternatives include the deployment of mobile store-and-forward nodes, or *data mules*, use of satellites (especially low-earth-orbiting satellites), or mobile routers taking the form of planes, balloons, etc.

These types of unusual interconnection technologies are characterized by different properties than are typical for Internet-like networks. As such, they are a form of *challenged networks* [1] that may require other approaches such as store-and-forward, hop-by-hop retransmission, and scheduled data communication. In particular, a network system for servicing sensor networks should be able to handle lack of infrastructure, scheduled connectivity, intermittency due to node or communication link failure, and significant heterogeneity among the various subnetworks used to provide service to the sensor network. Interestingly, many of the same problems occur *within* a sensor network. Although the timescales may be different, interference, high link error rates, scheduled node up time, and heterogeneity among sensor nodes is common and requires similar solutions.

In consideration of these issues, we suggest the recently-proposed Delay Tolerant Networking architecture [1] as an approach for solving such problems. The DTN architecture supports data mules and scheduled network connectivity, enhances end-to-end reliability with a hop-by-hop store and for-

ward mechanism, and accommodates heterogeneity by using a flexible naming and addressing scheme.

## 2. SENSOR NETWORKING CHALLENGES

In many typical configurations, each sensor relays its data through its neighbors to a gateway, which operates as the single point-of-exit. Routing protocols typically direct data from multiple sensors to the exit point in a many-to-one fashion [8]. When infrastructure is present, there is no particular difficulty in connecting the gateways to another infrastructure network such as the Internet. When no such infrastructure is available, this problem is much more difficult. We now focus on these problems in more detail:

- **network infrastructure:** Lack of infrastructure may force sensor network gateways to be intermittently connected to the operator's network. If the sensors generate data more frequently than connectivity becomes available, data must be buffered somewhere in the sensor network to avoid loss.
- **interruption:** Scheduled down time, interference, or environmental hostility may cause the interruption of otherwise-operable communication links. When scarcity of power makes communication costly and therefore infrequent, achieving efficient utilization of communication opportunities becomes very important.
- **heterogeneity:** Challenged networks cannot generally be assumed to be running a common set of protocols [1] in each node, thereby requiring some additional mechanisms to support interoperable communication. Any such approach will need to accommodate a high degree of variation in naming, addressing, rate control, and routing approaches. In particular, support for proxies that can be placed at convenient points of interconnection is of significant importance.

As an architecture for challenged internets, DTN incorporates many features that can be of great use in addressing these problems. Some of these mechanisms, such as in-network buffering, have already been employed unilaterally, to meet needs of specific sensor networks. We believe it would be better, however, to standardize on a complete architecture (and perhaps select an appropriate subset for specific implementations), instead of building each feature independently for each deployment. In the next sections, we present an overview of the relevant parts of the DTN architecture and a brief discussion of some sensor network deployment scenarios could benefit from adopting the DTN architecture.

## 3. OVERVIEW OF DTN ARCHITECTURE

The DTN architecture builds upon the abstraction of reliable asynchronous communication of application-specified messages. Facilities within its design include support for multipath routing, hop-by-hop reliability and retransmission, a

**Figure 1: Transmission of data from a sensor node to a destination application on the Internet using DTN. Since there is no contemporaneous path between the sensors and the Internet, the DTN bundle protocol stores the bundles until the next available connection arrives, allowing eventual delivery of the sensor data to its destination.**

set of *convergence layer protocols* which provide adaptation among underlying protocols, and a naming scheme that uses *late binding*. Late binding allows name-to-address mapping to be executed topologically near the target of a communication as opposed to the *early* binding performed by the DNS operation typically executed by today’s Internet applications. Here we expand on these features in the context of sensor networks.

### 3.1 Asynchronous Message Delivery

Multihop sensor networks often employ a limited amount of buffering, requiring data to be held for some time until disrupted connections are restored. However, even with this capability, the sensors must have a communication range twice that of the sensing range [4], resulting in higher equipment cost and more power consumption. Nodes closer to the gateway typically experience significant resource contention, as they relay data from other nodes in addition to themselves.

DTN’s asynchronous, store-and-forward message delivery model addresses the problems of interruptions and lack of infrastructure. Because DTN applications are generally tolerant to delay, data may be stored in a queue for long periods, if necessary. This flexibility allows the network to cope with a lack of available neighbors (either external to or internal within the sensor network) caused by interference, congestion, low-duty power cycling or lack of infrastructure. In extreme cases, where the sensor network is isolated, queuing may be for very long periods, including the time required for the next arrival of a datamule [7]. The use of mules can help to conserve sensor node battery power by reducing the distance data needs to be transmitted [4].

### 3.2 Fragmentation and Routing

The DTN architecture supports the development of sophisticated approaches to routing, well beyond typical shortest-path [2]. Using store-and-forward, DTN routing computations generally take place over a time-evolving graph where a source and sink may never have a contemporaneous end-to-end path available. The architecture supports *proactive* and

*reactive* fragmentation for handling interruptions. In proactive fragmentation, if communication is scheduled, queued messages can be split into appropriate-sized segments ahead of time and when a communication opportunity becomes available, exactly the correct quantity of data is transferred. When unexpected failures are encountered, DTN employs reactive fragmentation, which essentially amounts to re-packaging data received across a link so that it may be delivered as an independent fragment. Fragments are eventually re-assembled by the final receiver. The combination of these features, in conjunction with multi-path routing, provides better utilization of scarce communication resources.

### 3.3 Naming and Convergence Layers

Some sensor networks are internally heterogeneous. That is, they comprise nodes of more than one type and/or functionality [4, 5]. The basic motivation is to engineer the network such that more frequently-used sensor nodes will be provisioned with more power reserves, leading to more uniform sensor node lifetimes. However, despite the potential benefits, heterogenous sensor networks are less common because they require a more complicated architecture and supporting protocols.

Because of the additional complications in designing protocols for heterogeneous networks, many providers of sensor networks only support a single node architecture or protocol stack. Since the vendors each have their own communication protocols and naming schemes, sensors are frequently unable to inter-communicate, requiring each type of sensor to have its own separate infrastructure for data collection and delivery. The DTN addresses this problem with two mechanisms: a naming scheme capable of embedding the names/addresses used by other protocol families into its own names, and a *convergence layer* abstraction that provides a standardized way to adapt existing protocols to be used as underlay protocols for hop-by-hop DTN message delivery. These concepts apply equally well inside or among sensor networks.

DTN’s flexible naming scheme uses a tuple consisting of a globally unique region identifier, which acts as a routing hint, and a region-specific administrative ID. Both are variable-length. The administrative ID is only resolved (if required by the underlying protocols) inside the destination region of interest, thereby providing late binding capability. It is treated as an opaque value by DTN routing otherwise. This structure allows for the re-use of node administrative identifiers in different target regions.

The convergence layer abstraction helps to solve the problem of adapting different underlying protocols (transport or otherwise) to be useful for supporting the DTN delivery protocol (bundle delivery). A convergence layer for a specific protocol adds functionality to support reliable delivery of DTN messages over the corresponding transport layer (e.g. augmenting TCP with message boundaries, and UDP with reliability and sequencing). Convergence layers also handle the signaling for fragmentation and connection re-establishment, if appropriate. Overall, a DTN router acts as a form of proxy among differing network types and is agnostic regarding the protocol layering employed in the networks it interconnects.

## 4. DEPLOYMENT SCENARIOS

In this section we present several deployment scenarios for sensor networks, and show how the DTN architecture can be applied to enable interoperability and reliability.

**Figure 2: The Packbot travels through the sensors, using the DTN infrastructure to collect the data.**

## 4.1 ZebraNet

ZebraNet [3] uses custom tracking collars in a peer-to-peer network to track wildlife in Kenya. Each collar collects a zebra's mobility pattern and communicates the data in an epidemic fashion to other collars and a mobile base station that is only available when researchers are in the field. The system uses store-and-forward asynchronous messaging and pays careful attention to scheduling communications in order to conserve battery life.<sup>1</sup>

Network connectivity between the zebras and the base station (and between the zebras themselves) is intermittent and opportunistic. These features, combined with the fact that the base station may not yet be present when the data is collected and distributed, necessitates the use of epidemic routing and network storage. In addition, connectivity between the base station and the Internet is not maintained out in the field. Thus, the base station must store the data until its connectivity to the Internet can be re-established.

Currently, these operations happen manually - data is collected in the field and then manually uploaded to the Internet by the researchers. However, this requires the periodic presence of experts on site. Instead, the base station could be configured as a DTN data mule, which can physically transport the data from the field to the Internet. Establishment of connectivity with the zebra network results in automatic transfer of the pending data bundles to the base station, and subsequent re-establishment of connectivity with the Internet delivers the data to the researchers, regardless of location. In this case, the only personnel required in the field is the driver of the vehicle carrying the base station.

## 4.2 Ad Hoc Seismic Array

Packbot is an unmanned ground vehicle (UGV) developed by the US military and used by researchers at UCLA as a data mule for collecting data in their heterogeneous sensor network [4]. It carries a mobile *microserver* (embedded Linux system), carried through the environment to collect data from widely distributed stationary sensors. It is designed to support a long term deployment (50-100 nodes, each 5-10km apart, stretching for 500km through Mexico), but also supports groups of nodes separated by long distances [6].

<sup>1</sup>GPS units present in each collar provide location and the timing synchronization for scheduling communications.

This deployment employs a multihop solution - messages containing sensor data may make several hops to reach the path of the data mule. So, each node selects the best (sensor node) next hop based on buffer availability and proximity to the packbot path. There are no end-to-end acknowledgements - each hop is a reliable transfer. Once a message reaches the packbot, it is buffered until the packbot reaches the final stationary gateway on its path, where it is delivered across the Internet.

The DTN architecture serves as an appropriate framework for this diverse scenario. In an environment where end-to-end paths are often not available, for example, when the packbot is not currently present, the DTN architecture supports buffer management in addition to multihop delivery of data. DTN data messages travel as far as the path is available, and are buffered until the data mule arrives. If the path and schedule of the packbot is known in advance with some degree of precision, low-duty power cycling can be accommodated as well.

## 5. CONCLUSION

In this paper, we presented the types of challenges encountered in constructing and deploying sensor networks, a DTN-based approach to dealing with them, and a brief description of some existing sensor networks that could benefit from the architecture. The DTN architecture, with its explicit support for store-and-forward data routing and a flexible naming approach that accommodates radical heterogeneity, can contribute both directly in connecting remote sensor networks to infrastructure networks (e.g. the Internet), as well as indirectly within sensor networks themselves. In the later case, we might expect an interoperable subset of the overall DTN ideas to be implemented inside sensor networks, particularly those composed of nodes with limited memory and communication resources. In effect, the DTN architecture provides a common framework and standardized approach to the interconnection of networks that suffer frequent disruption, such as sensor networks, and offers the benefits of improved interoperability and avoidance of duplicate effort in solving the problems of network disruption.

## 6. REFERENCES

- [1] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proc. SIGCOMM 2003*, Aug. 2003.
- [2] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. SIGCOMM 2004*, Aug. 2004.
- [3] P. Juang, H. Oki, Y. Wang, et al. Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet. In *Proceedings of ASPLOS-X*, Oct. 2002.
- [4] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Strivastava, and D. Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of ACM MobiSys*, June 2004.
- [5] V. Mhatre and C. Rosenberg. Homogeneous vs. heterogeneous clustered sensor networks: A comparative study. In *Proceedings of IEEE ICC*, June 2004.
- [6] A. Parker, A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Strivastava, and D. Estrin. UCLA DTN Sensor Networks Update, Aug. 2004. Available at <http://www.dtnrg.org>.
- [7] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. In *Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications*, May 2003.
- [8] A. Woo, T. Tong, and D. Culler. Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks. In *Proc. SenSys*, pages 14-27, 2003.